



INSTITUTO FEDERAL

Catarinense

Campus Camboriú

Matrizes

Professora: Lissandra Maiara Fischer

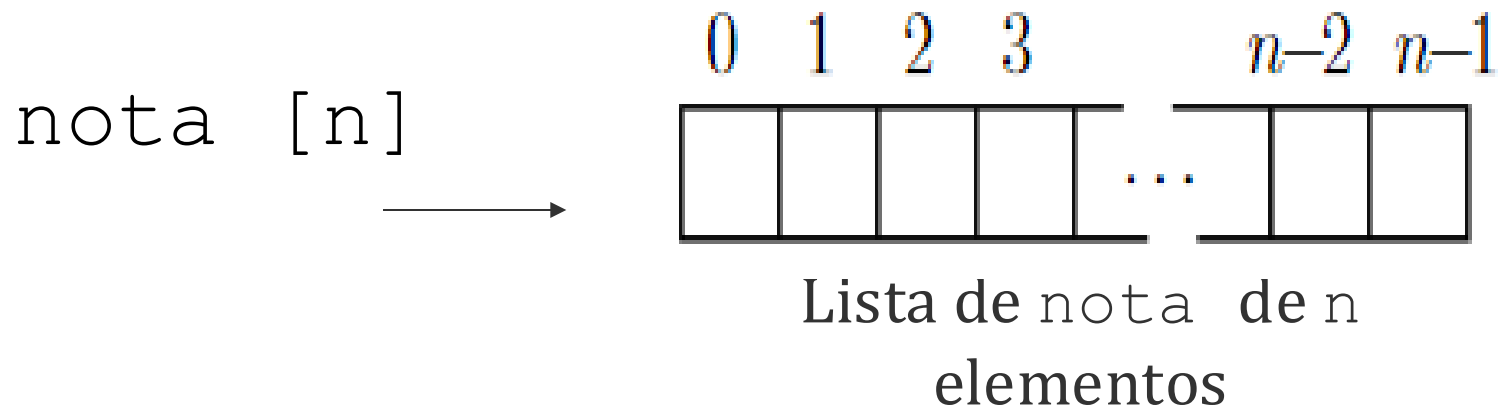
Objetivo

- Compreender o conceito da **estrutura de dados** conhecida como **matrizes** (em Python).
 - declaração
 - inicialização
 - atribuição
 - acesso ao conteúdo

Como representar uma matriz?

- Array **unidimensional**

- Uma **listas** identificada por *nota* de *n* posições:



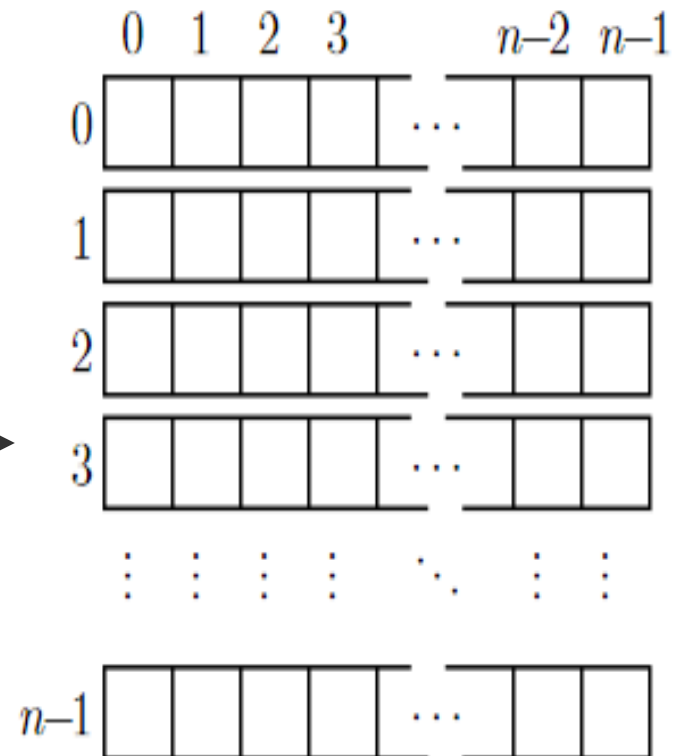
Como representar uma matriz?

- Listas de Listas é uma estrutura **bidimensional**, ou seja **matriz**
 - Uma lista identificada por **nota** de **$n \times n$** posições:

nota [n] [n]

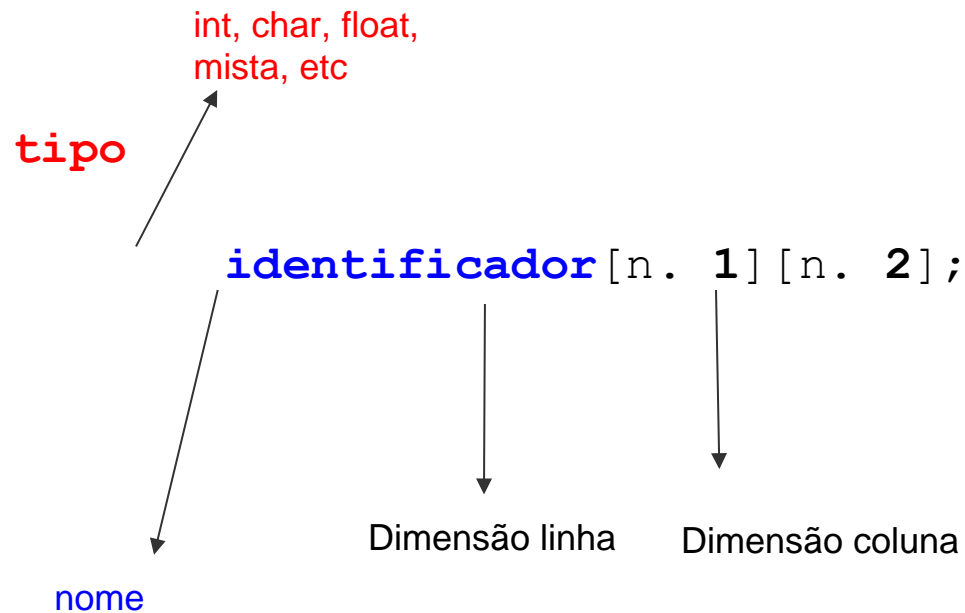


Lista de listas
nota de $n \times n$
elementos



Matrizes

● Em Python:



Exemplos

Matriz

| | 0 | 1 | 2 | 3 | 4 | 5 |
|-----|---|---|---|---|---|---|
| x 0 | | | | | | |
| 1 | | | | | 5 | |



Acessando elementos
em Python

`x[1][4]`

MAT

| | 0 | 1 | 2 |
|---|---|---|---|
| 0 | | | |
| 1 | | | |
| 2 | | | |
| 3 | | | |



Declarando listas em
Python

`MAT = [[1, 2, 3], [4, 5, 6], [7, 8, 9], [10, 11, 12]]`

Exemplos

Matriz

| | | 1 | 2 | 3 | 4 | 5 |
|---|---|----|----|---|----|-----|
| X | 1 | 12 | 9 | 3 | 7 | -23 |
| | 2 | 15 | 4 | 2 | 34 | -4 |
| | 3 | 3 | 45 | 3 | 0 | -3 |



Declarando listas em
Python

```
X = [[12, 9, 3, 7, -23],  
      [15, 4, 2, 34, -4],  
      [3, 45, 3, 0, -3]]
```

| | | 0 | 1 | 2 |
|-----|---|---|---|---|
| MAT | 0 | | | |
| | 1 | | | |
| | 2 | | | |
| | 3 | | | D |



Acessando elementos
em Python

```
MAT[3][2]
```

Exemplos

Acessando elementos em Python

```
X[3][4] = 0;  
X[2][1] = -5;  
X[4][4] = 1;
```



| | 0 | 1 | 2 | 3 | 4 |
|---|---|----|---|---|---|
| 0 | | | | | |
| 1 | | | | | |
| 2 | | -5 | | | |
| 3 | | | | | 0 |
| 4 | | | | | 1 |

X

Matriz

```
uma_lista = [3, 67, "gato",  
             [56, 57, "cachorro"], [ ],  
             3.14, False]
```

```
uma_lista[5] = 3.14  
uma_lista[3][0] = 56  
uma_lista[3][2][0] = c  
uma_lista[2][0] = g
```


Como declarar uma matriz (lista de listas)?

```
matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
frutas = ["maca", "pera", "jaca", "caju"]
```

| | frutas | | | |
|---|--------|---|---|---|
| | 0 | 1 | 2 | 3 |
| 0 | m | a | c | a |
| 1 | p | e | r | a |
| 2 | j | a | c | a |
| 3 | c | a | j | u |

Podemos utilizar numpy?

```
import numpy as np
```

```
M1 = np.array([[1,2],[3,4]])
```

```
print(M1)
```

```
[[1 2]
 [3 4]]
```

Sim! Desde que os dados a serem armazenados sejam apenas numéricos.

Como preencher uma matriz

```
import numpy as np

# define o tamanho do array
N = 3

#preenche a estrutura com zeros
matriz = np.zeros((N,N))

#mostra a matriz
print(matriz)
```

```
[[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]
```

Preencher uma matriz usando listas

```
matriz = [[0, 0, 0],[0, 0, 0], [0, 0, 0]]

# Inserindo dados na matriz usando for.
for l in range(len(matriz)):
    for c in range(len(matriz[l])):
        matriz[l][c] = int(input("Informe valores
                                para a matriz"))

print("Matriz 3x3: ", matriz)
```

Preencher uma matriz

```
l = 0
# Inserindo dados na matriz usando while.
matriz = [[0, 0, 0],[0, 0, 0], [0, 0, 0]]
while l < len(matriz):
    c = 0
    while c < len(matriz[l]):
        matriz[l][c]= int(input("Informe valores para a
matriz"))
        c +=1
    print()
    l += 1

print("Matriz 3x3: ", matriz)
```

Preencher uma matriz

```
matriz = []

# Inserindo dados na matriz usando for através da
linha.
for l in range(3):
    lista = []
    for c in range(3):
        lista.append(int(input("Informe valores
                                para a matriz")))
    matriz.append(lista)

print("Matriz 3x3: ", matriz)
```

Exibir uma matriz

```
matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
print("Matriz 3x3: ", matriz)
```

```
Matriz 3x3: [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
# Exibindo a matriz com for.
```

```
print("Matriz impressa de outra forma:")
```

```
for lista in matriz:
    for elemento in lista:
        print(elemento, end=' ')
    print()
```

```
1 2 3
4 5 6
7 8 9
```

```
l = 0
```

```
# Exibindo a matriz com while.
```

```
print("Matriz impressa de outra forma:")
```

```
while l < len(matriz):
    c = 0
    while c < len(matriz[l]):
        print(matriz[l][c], end=' ')
        c += 1
    print()
    l += 1
```

```
1 2 3
4 5 6
7 8 9
```

Referências



Referências Básicas

FORBELLONE, André Luiz Villar; EBERSPÄCHER, Henri Frederico. **Lógica de programação: a construção de algoritmos e estruturas de dados**. 3. ed. Pearson Prentice Hall. 2005

MANZANO, José Augusto N. G; OLIVEIRA, Jayr Figueiredo de.. **Algoritmos: lógica para desenvolvimento de programação de computadores..** 27. ed.. Érica. 2014

Referências Complementares

DOWNEY, Allen B. **Pense em Python**. 2ª Ed. Novatec. 2016

MENEZES, Nilo Ney de Coutinho. **Introdução a programação com Python**. 3ª Ed. Novatec. 2019

CORMEN, Thomas H et al. **Algoritmos: teoria e prática**. 2. ed. Elsevier, Campus,. 2002

Referências na Internet

<https://docs.python.org/3/>

<https://www.w3schools.com/python/default.asp>